Query Optimization in Context of Pseudo Relevant Documents

Ashish Kishor Bindal[1], Sudip Sanyal[1]

[1] Indian Institute of Information Technology, 211012 Allahabad (U.P.), India
{ashish.bindal@epfl.ch, ssanyal@iiita.ac.in}

Abstract. In conventional vector space model for information retrieval, query vector generation is imperfect for retrieval of precise documents which are desired by user. In this paper, we present a stochastic based approach for optimizing query vector without user involvement. We explore the document search space using particle swarm optimization and exploit the search space of possible relevant and non-relevant documents for adaption of query vector. Proposed method improves the retrieval accuracy by optimizing the query vector which is generated in conventional vector space model based on various term weighting strategies including TF-IDF and document length normalization. Our experimental result on two collections Medline and Cranfield shows that adapted query vector in pseudo relevant document performs better over the classical vector space model. We achieved improvement of 3-4% in Mean Average Precision (MAP) and 5-10% in Precision at lower recall. Further expansion of search space in pseudo non-relevant documents does not lead to significant improvement, but proper representation of pseudo non-relevant document leaves a scope in future to guide the better optimization of query vector.

Keywords: Information Retrieval, Pseudo Relevance Feedback, Particle Swarm Optimization.

1 Introduction

The explosive growths of information sources and Internet have had impact on the rapid growth of repositories of textual data. Therefore the role of Information Retrieval System (IRS) has become significantly important to retrieve more precise result from large information collection in response to queries fired by users.

For more than three decades, query reformulation has been the primary research focus in domain of information retrieval. There are two query reformulation approaches: query expansion [7][8][9][10][19] and query reweighting [8][17][18]. These two approaches can be further classified into four categories; (i) Techniques based on manual thesaurus [19]; (ii) Based on information extracted from collection of documents [10] [8]; (iii) Based on user feedback information about relevant and irrelevant documents (relevance feedback) [5][6][18]. (iv) Based on Pseudo Relevance Feedback (PRF) [20][21].

Robertson [5] and Salton [6] use relevance feedback to modify the query by taking into account the user relevance feedback on the documents retrieved by original query. In [8], query terms are reweighted to incorporate the term relatedness using statistics of term co-occurrence in document collection. In [22], Rila Mandala expands the query where expansion terms are taken from manual thesaurus and automatically constructed thesaurus.

Recently Machine learning approaches have attracted attention of researchers in Information Retrieval. Kraft [18] proposed the use of genetic algorithm (GA) to select the best query term for query expansion. In [19], Lourdes Araujo employs genetic algorithm to select the new candidate terms, provided by a morphological thesaurus, for the query expansion. This system uses the relevance feedback from users to change representations of authors, index terms and documents over time. In [1] Ziqiang Wang applied the particle swarm optimization algorithm (PSO) for query reweighting using information derived from user feedback. It shows substantial improvement in precision of IRS over genetic algorithm and relevance feedback. In [2] a hybrid GA-PSO based algorithm is used for query expansion with relevance feedback.

A common factor of the above mentioned work [1][2] is that they are based on feedback provided by the user. Past research has verified the effectiveness of relevance feedback. However, users are often lazy to provide relevance judgments. In this work we propose a new application of PSO to the Query optimization without user intervention. Assuming that top-k retrieved documents are all relevant, referred as pseudo Relevant Document, our proposed method searches the best combination of weights of query terms to improve the recall and precision of IRS. Along with this assumption, we have also experimented with the information from retrieved documents which are far away from the original query, referred as pseudo non-relevant document, to further improve the efficiency of IRS. The rest of the paper is organized as follows. In section 2, we briefly review the particle swarm optimization algorithm. In section 3, we present an application of PSO for query reweighting. In section 4, we show the experiment analysis on Medline and Cranfield dataset. The conclusions are discussed in section 5.

2 Particle Swarm Optimization

Particle Swarm Optimization (PSO), developed by Kennedy and Elberhart in 1995, is a population based swarm intelligence technique [12]. PSO simulates the social behavior of birds flocking as an evolution criterion. Individuals or particles are evolved by cooperation and competition among themselves to discover the possible best solution in a given search space.

PSO is initialized with a group of random particles as candidate solutions in an n-dimensional space, where position of i^{th} particle in the group is represented as $\mathbf{X}_i = <x_{i1}, x_{i2}, x_{i3}\ldots x_{in}>$. Fitness function evaluates the position in n-dimensional search space to evaluate the fitness of particle. Each particle keeps track of the best solution it has achieved so far. This value is called personal best (*pid*) and represented as

pid=<p_{i1}, p_{i2}… p_{in}>. Moreover, best location achieved among particles, is called as Global Best Position (*pgd*). These two best positions decide the direction of particle's movement according to equation (1). During evolution a particle continuously adjusts its direction toward best known solution for flying towards a new position according to equation (2). Each particle has following attributes: current position ($\boldsymbol{X}_i$), personal best position (*pid*), global best position (*pgd*) and velocity (V_i). Initially each particle is assigned a random position ($\boldsymbol{X}_i$), velocity (V_i) and personal best position (*pid*).

$$Vi(t+1) = K[wi * \ Vi(t) + \ c1 * r1 * (pid - Xi) + c2 * r2 * (pgd - xi)] \quad (1)$$

$$Xi(t+1) = \ Xi(t) + \ Vi(t+1) \quad (2)$$

Where *r1* and *r2* are random number between [0,1], *t* denotes the epoch number and positive acceleration constants that pull each particle towards pid and pgd are represented by *c1* and *c2*. *wi* is the inertia weight to provide balance between global and local exploration [16]. Where Clerk's constriction factor (K) prevents the system explosion and insures the convergence of particle's system [13].

Unlike GA, information sharing among individuals in PSO is one way which is through *pgd* only. Hence all individuals tend to converge to best solution.

3 PSO Model for Query Optimization

Vector space model is used as the underlying framework in which document and query are represented in the vector space as vectors. The terms in the documents and query are assigned weights in classical model on the basis of its frequency in the document and the inverse document frequency. However weight of the term also depends on its importance in the context which is specified by query. The goal of PSO is to learn the significance of query term in the form of weight from the context provided by the documents. PSO algorithm incorporates top-k retrieved documents and pseudo non-relevant document for finding optimal query vector to improve the effectiveness of IRS. In the following subsections we explain how PSO can be used to improve the weights of the terms for a given query.

3.1 Particle Swarm Optimization Steps

STEP 1:- Encoding of query: Initial step in PSO is the definition of particle to be optimized. In this paper, particle is represented by query vector. Each particle representing the query is of the form

$$Qu = (qu_1, qu_2, qu_3, \ldots, qu_T)$$

Where *T* is the number of stemmed terms in the query; qu_i defines the importance of term the i^{th} in the query. Initially term weights are assigned either randomly or

through some query weighting scheme. These weights are then evolved through generations. We used the following query weighting formula [15]:

$$qui = \frac{(1+log(tfui)\ )*\log\left(\frac{N}{ni}\right)}{\sqrt{\sum_{j=1}^{T}\left((1+log\ (tfuj))*log\left(\frac{N}{nj}\right)\right)^2}} \quad (3)$$

Where, *tfui* is the term frequency of *qui* term in query, $log(N/n_i)$ is the inverse document frequency, *N* is the number of total documents and *nj* is the number of documents containing the term qu_i. In our approach, we generated an initial population which contains few particles of initial population with above weighting scheme to give good direction for evolution and the rest of them with random weights to explore different random regions of problem space.

STEP 2:- Fitness is assigned to each individual in the population. This represents the effectiveness of a query vector during the retrieval stage. Fitness function that we have proposed uses the degree of similarity between query, and top-k document which are retrieved from original query vector. To compute this similarity, we have employed the vector space model of information retrieval. In this model, a document d_j and *Qu* are represented as *T*-dimension vector. To construct the vector, we have to assign the weights to stemmed terms left after removing stop words from both document and query. Weight of the terms in document vector is computed using equation (3). The degree of similarity of the document d_j with respect to query *Qu* is evaluated as the cosine of the angle between these two vectors using following equation.

$$Sim(dj, Qu) = \frac{\sum_{i=1}^{T} qui*dji}{|dj|*|Qu|} \quad (4)$$

Where $|d_j|$ and $|Qu|$ are the document and query vector length respectively. Fitness function is computed according to the similarity measure of top-*k* retrieved documents. The formula is:

$$F1(Qu') = \frac{\sum_{i=1}^{K} Sim(di,Qu)}{K} \quad (5)$$

Where *Qu* is the original query vector, *k* is the number of top retrieved documents which are assumed as all relevant documents. This fitness function would favor the query evolution in pseudo relevant documents. To improve the effectiveness of IRS, we examined the similarity measure of documents which are far away from query (pseudo non-relevant documents). Fitness function for the same is:-

$$F2(Qu') = \frac{\sum_{i=1}^{K} Sim(di,Qu)}{K} - \frac{\sum_{i=s1}^{S2} Sim(di,Qu)}{|s2-s1|} \quad (6)$$

Where [*S1, S2*] define the rank of retrieved documents from original query. Documents lying in the range of [*S1,S2*] are considered as pseudo non-relevant documents.

STEP 3:- Each particle compares its fitness value evaluated using equation (5) or (6) with particle *pbest* fitness value. If current value is greater than *pbest*, then *pbest*

value is updated with current particle position in T-dimensional space. In first iteration, particle current position is set as *pbest* position. Each particle compares its fitness value with global best value *pgb*. If current value is better than *gbest* then reset *gbest* with current location.

STEP4:- Change the velocity and position of the particle according to equation (1) and (2).

STEP 5:- Until a termination criteria is met, loop to step 2. Termination criteria are either maximum number of iterations or till adequate amount of updation is achieved in fixed max number of iteration. Whichever of these two conditions is satisfied, termination criteria is satisfied.

STEP 6:- Global best position in final iteration of PSO in T-dimensional search space is considered as optimized query vector. Rank the document on the basis of cosine similarity function (4) using optimized query vector.

STEP 7:- End.

3.2 Parameter Control

The advantages with PSO algorithm are easy implementation, fast convergence and tuning of few parameters. The parameters and their values, which are used for analyzing and carrying out the experiment, are as follows:-

Population size is kept fixed of 20 particles. Range for each dimension of particle is fixed to [0, 1]. In case of any dimension crossing the range from right side then it's kept at max value i.e. 1. Value of both accelerating factors *c1* and *c2* is usually set to 2 in the PSO. Here, *c1* and *c2* control how far a particle will move in a single iteration. The inertia weight *w* in equation (7) is also used to balance between global best and personal best. PSO adapts the value of *w* such that it linearly decreases from 0.9 to 0.6 over the generations.

$$wi = wmax - \frac{wmax - wmin}{iter\_max} * iter \tag{7}$$

Where *wmax*=0.9 and *wmin*=0.6, *iter* is the current iteration and *iter_max* is the maximum number of iterations. In order to prevent the system explosion and to insure the convergence of the PSO algorithm, the Clerk's constriction factor *K* is defined as follows:-

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^4 - 4*\varphi}\right|} \tag{8}$$

Where $\varphi = c1+c2$ and $\varphi > 4$.

Top 5 retrieved documents are assumed as relevant documents for evaluation of fitness of query using fitness function *F1*. For Fitness function *F2*, value of *k* remains same and range [*S1*, *S2*] for pseudo non-relevant document is found empirically for both Medline and Cranfield collection.

4 Experimental Result

In order to evaluate the performance of proposed method, we used two document collections Medline and Cranfield. Medline contains 1033 documents and 30 queries whereas Cranfield contains 1400 document and 225 queries.

For experimental purpose, we employed a vector space model with improved version for comparing against the performance of proposed method. We removed stopwords from query and documents and then stemmed the remaining words using the Porter stemmer. Documents and query are presented as vectors where the term weight used for both document and query in the initial retrieval of document is done using equation (3). Ranking of documents for the query is done by similarity function in equation (4). Finally the system calculates precision and mean average precision (*MAP*) for comparing performance of proposed approach.

Table 1. Result on Medline dataset for Fitness function (F1)

P@	Original Query	The Proposed Method (F1)	Improvement
Map	0.536	0.55888	4.21%
P@5	0.687	0.753	9.70%
P@10	0.647	0.682	5.56%
P@15	0.598	0.622	4.08%
P@20	0.545	0.571	4.82%

Table 2. Result on Cranfield dataset for Fitness function (F1)

P@	Original Query	The Proposed Method (F1)	Improvement
Map	0.4104	0.4233	3.13%
P@5	0.4329	0.4442	2.61%
P@10	0.2973	0.3102	4.34%
P@15	0.2388	0.2465	3.24%
P@20	0.1953	0.2038	4.17%

4.1 Effect of Pseudo Relevant Document

Table 1 makes a comparison of the original query vector and reweighted query vector in context of pseudo relevant documents (F1) with the use of mean average precision and the average precision on Medline dataset. Precision is defined as follows:

$$Precision\ Rate = \frac{|Ra|}{|A|} \tag{9}$$

Where, *Ra* is number of correct results and *A* is number of all returned results. Similarly Table 2 shows a comparison on Cranfield dataset. We can see that our approach gives improvement on both the measures for both the collection over classical vector space model.

Table 3. Result on Medline dataset for Fitness function (F2)

P@	The Proposed Method (F1)	The Proposed Method (F2)	Improvement
Map	0.5589	0.5688	1.79%
P@5	0.753	0.728	-3.32%
P@10	0.682	0.680	-0.39%
P@15	0.622	0.640	2.86%
P@20	0.571	0.582	1.81%

Table 4. Result on Cranfield dataset for Fitness function (F2)

P@	The Proposed Method (F1)	The Proposed Method (F2)	Improvement
Map	0.4233	0.4277	1.03%
P@5	0.444	0.435	-2.058 %
P@10	0.310	0.305	-1.74%
P@15	0.247	0.252	2.27 %
P@20	0.204	0.207	1.66 %

4.2 Effect of Pseudo Non-Relevant Documents

Table 3 and 4 compare the performance of pseudo relevant documents based fitness function (*F1*), and *F2* which exploit the information from pseudo non-relevant documents. Range of pseudo non-relevant documents [*S1*,*S2*] for *F2* is determined empirically. Each individual data item has been computed as the average over 5 different runs. Improvement in *MAP* is marginal on the Cranfield and Medline collections. Precision at 5 docs and 10 docs decreases as compared to previous hypothesis (*F1*). However, we can see the improvement in precision at 15 and 20 docs though the improvement is not substantial overall.

Table 5. Values of [S1, S2], for Cranfield and Medline dataset

Range	Medline	Cranfield
S1	60	80
S2	75	90

5 Conclusion

In this paper we have shown how a stochastic algorithm can help to optimize the query vector to improve the efficiency of IRS using pseudo relevant documents. Our method does not require any feedback from user. Specifically, we have assumed top-*k* retrieved document from original query vector as all relevant document. The particle swarm optimization algorithm chooses the appropriate combination of weights of terms in query vector and uses fitness function as a measure of the proximity between the reweighted query vector and top-*k* ranked documents. The proposed method increases the precision rate and the *MAP* significantly for both the collections. In Medline, improvement in Precision at 5 (P@5) is more than Precision at 10 (P@10), while in Cranfield, improvement in P@10 is more as compared to improvement at P@5. This concludes that if proposed method does not increase the precision at low value of *n* (P@n) then it increases the improvement in precision at high value of *n1* (P@n1) significantly, where (*n1*>*n*). Moreover, there is improvement in comparison to traditional Vector Space Model at all values of *n*. It is evident that optimized query vector improves the ranking of retrieved documents corresponding to original query vector and includes new relevant documents.

Along with the assumption of pseudo relevant document, we have also investigated for negative relevance feedback using pseudo non-relevant documents. Precision at lower recall (P@5, P@10) decreases due to the inclusion of pseudo non-relevant document. Good precision at top ranked document is crucial from the perspective of the user. Finding optimal range of pseudo non-relevant document is more difficult to accomplish than for pseudo relevant document. Nevertheless, there is marginal improvement in overall mean average precision and at higher recalls. Query vector is unable to learn from pseudo non-relevant document due to missing of query terms altogether at lower rank. An interesting future research direction to improve the query adaption is to find range of pseudo non-relevant document at lower recall for negative feedback such that it incorporates more relation between the query terms.

6 Acknowledgements

The authors are grateful to IIIT-A for providing infrastructural support.

7 References

1. Ibrahim, S. N., Selamat, A., and Selamat, M. H. 2009. Query Optimization in Relevance Feedback Using Hybrid GA-PSO for Effective Web Information Retrieval. In Proceedings of the 2009 Third Asia international Conference on Modeling & Simulation (May 25 - 29, 2009). AMS. IEEE Computer Society, Washington, DC, pp. 91-96.
2. Ziqiang Wang, Xin Li, Dexian Zhang, Feng Wu: A PSO-Based Web Document Query Optimization Algorithm. ASWC 2006: pp. 609-615
3. D. Grossman and O. Frieder, Information retrieval: Algorithms and heuristics. Springer, 2004

4. Salton, G. and Buckley, C. 1997. Improving retrieval performance by relevance feedback. In Readings in information Retrieval, K. Sparck Jones and P. Willett, Eds. Morgan Kaufmann Multimedia Information And Systems Series. Morgan Kaufmann Publishers, San Francisco, CA, pp. 355-364.
5. Robertson, S. E. and Sparck Jones, K. 1988. Relevance weighting of search terms. In Document Retrieval Systems, P. Willett, Ed. Taylor Graham Series In Foundations Of Information Science, vol. 3. Taylor Graham Publishing, London, UK, pp. 143-160.
6. Salton, G. 1971 The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall, Inc.
7. Y. C. Chang, S. M. Chen and C. J. Liau, A new query expansion method based on fuzzy rules, Proceedings of the 2003 Joint Conference on AI, Fuzzy System, and Grey System, Taipei, Taiwan, Republic of China, 2003.
8. B. M. Kim, J. Y. Kim and J. Kim, Query term expansion and reweighting using term co-occurrence similarity and fuzzy inference, Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, Canada, Vol.2, pp.715-720, 2001.
9. Y. Qiu and H. P. Frei, Concept based query expansion, Proceedings of the 13th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, pp. 160-169, 1993.
10. J. Xu and W. B. Croft, Query expansion using local and global document analysis, Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, Zurich, Switzerland, pp. 4-11, 1996.
11. Y. J. Horng, S. M. Chen and C. H. Lee, A new fuzzy information retrieval method based on document terms reweighting techniques, International Journal of Information and Management Sciences, Vol.14, No.4, pp. 63-82, 2003.
12. Eberhart, R.C., Kennedy, J.: A new Optimizer using particle swarm theory. In: Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43 (1995).
13. M. Clerc, J. Kennedy, The particle swarm: explosion stability and convergence in a multidimensional complex space, IEEE Trans. Evolution. Comput. 6 (1) (2002) pp. 58-73
14. Gerard Salton, A. Wong, and C. S. Yang. A vector space model for information retrieval. Communications of the ACM, 18(11): pp. 613–620, November 1975.
15. Singhal, A., Buckley, C., and Mitra, M. 1996. Pivoted document length normalization. In Proceedings of the 19th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Zurich, Switzerland, August 18 - 22, 1996). SIGIR '96. ACM, New York, NY, pp. 21-29.
16. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation, pp. 69-73. IEEE Press, Piscataway NJ (1998).
17. Y. J. Horng, S. M. Chen and C. H. Lee, A new fuzzy information retrieval method based on document terms reweighting techniques, International Journal of Information and Management Sciences, Vol.14, No.4, pp. 63-82, 2003.
18. D. H. Kraft, F. E. Petry, B. P. Buckles, and T. Sadasivan, "The use of genetic programming to build queries for information retrieval," in Proc. 1st IEEE Conf. Evol. Comput., vol. 1, 1994, pp. 468–473.
19. Araujo, L. and Pérez-Agüera, J. R. 2008. Improving query expansion with stemming terms: a new genetic algorithm approach. In Proceedings of the 8th European Conference on Evolutionary Computation in Combinatorial Optimization (Naples, Italy, March 26 -

28, 2008). J. Van Hemert and C. Cotta, Eds. Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, pp.182-193.
20. Chris Buckley, Gerard Salton, James Allan: Automatic Retrieval With Locality Information Using SMART. TREC 1992: pp. 59-72
21. Xu, J. and Croft, W. B. 1996. Query expansion using local and global document analysis. In Proceedings of the 19th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Zurich, Switzerland, August 18 - 22, 1996). SIGIR '96. ACM, New York, NY, pp. 4-11.
22. Mandala, R., Tokunaga, T., and Tanaka, H. 2000. Query expansion using heterogeneous thesauri. Information Processing & Management. 36, 3 (May. 2000), pp. 361-378.